

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**PROGRAMSKA PODRŠKA ZA REPRODUKCIJU
VIDEO SADRŽAJA ZA VERIFIKACIJU ADAS
ALGORITAMA NA STRANI PC RAČUNALA**

Diplomski rad

Ivana Škorić

Osijek, 2018.

Sadržaj

1. UVOD.....	1
2. AUTONOMNA VOŽNJA I NAČINI PRIKUPLJANJA PODATAKA ZA TESTIRANJE ADAS ALGORITAMA	3
2.1 ADAS sustavi	3
2.2 AMV ADAS GRABBER	5
2.3 Preduvjeti za dostavljanje sadržaja AMV ADAS Grabber uređaju	8
3. OPIS PROGRAMSKOG RJEŠENJA	10
3.1 Koncept rješenja	10
3.2 Čitanje iz datoteke	11
3.3 Rad s kontrolnim porukama	12
3.3.1 Zaglavlje datoteke.....	12
3.3.2 Zaglavlje okvira.....	16
3.4 Smještanje u memoriju	17
3.5 Prebacivanje podataka iz memorije PC-a u memoriju ploče korištenjem DMA	21
4. TESTIRANJE PROGRAMSKE PODRŠKE ZA REPRODUKCIJU VIDEO SADRŽAJA ZA VERIFIKACIJU ADAS ALGORITAMA NA STRANI PC RAČUNALA	24
4.1 Testiranje brzine prijenosa podataka preko PCIe sabirnice	24
4.2 Testiranje gubitka paketa.....	26
4.3 Testiranje brzine čitanja podataka s tvrdog diska.....	26
5. ZAKLJUČAK.....	29
LITERATURA	30
SAŽETAK	31
ABSTRACT	32
ŽIVOTOPIS	33

<i>PRILOZI</i>	34
----------------------	----

1. UVOD

U posljednje vrijeme došlo je do značajnog napretka u automobilske industriji glede autonomne vožnje, te će u budućnosti to biti jedna od ključnih uloga u inteligentnim transportnim sustavima. Kako bi se ta vozila implementirala u stvarnom svijetu, tj. na stvarnim cestama i u svakodnevnom prometu, oni moraju biti u stanju samostalno voziti prometnim putevima a da pri tome ne ugrožavaju sigurnost ostalih sudionika u prometu. Naime, autonomna vozila pomoću različitih računalnih sustava donose odluke prilikom vožnje te su s tim ciljem razvijani brojni algoritmi koji će to omogućiti. Do svega navedenog dolazi prije desetak godina, kada stručnjaci uviđaju koliko se nesreća događa zbog ljudske pogreške te pronalaze nova rješenja u vidu različitih računalnih algoritama koji vozaču asistiraju prilikom vožnje, poznatijih kao ADAS (engl. *Advanced Driver Assistance System*) algoritmi.

Danas su autonomna vozila dio svakodnevnice, a razina autonomije se povećava iz dana u dan. Algoritmi koji se razvijaju obrađuju informacije dobivene iz okoline koje dobivaju preko različitih senzora, kao što je primjerice kamera. Budući da vozilo donosi odluke na osnovu onoga što se nalazi ispred njega u stvarnom svijetu, jednu od najvažnijih uloga predstavljaju informacije dobivene s kamere. Jedan takav primjer je detekcija objekata ispred vozila koja se zasniva na računalnim algoritmima za detekciju i klasifikaciju objekata koji se nalaze ispred vozila. Nakon što su objekti detektirani i klasificirani, primjenjuje se algoritam koji odlučuje kakvu će akciju vozilo napraviti (kočenje, zaobilaženje, usporavanje i sl.) u situaciji kada se ispred njega nalazi detektirani objekt. Još jedan primjer implementacije ADAS-a podrazumijeva razvoj algoritma koji će omogućiti vozaču pomoć pri parkiranju. U ovom slučaju potrebno je koristiti kamere koje omogućuju pregled okoline vozila sa stražnje strane i odozgor, tako da će korisnik vozila moći vidjeti prepreke koje možda nisu odmah vidljive. Iz svega navedenog, zaključuje se kako ADAS sustavi predstavljaju veliki preokret u automobilske industriji i kako donose nove mogućnosti koje će podići razinu sigurnosti na cestama i olakšati ljudsku svakodnevnicu.

ADAS algoritmi obično se zasnivaju na strojnom učenju (engl. *machine learning*). Kako bi se postigla visoka preciznost, strojno učenje zahtijeva velike količine podataka posebno u slučaju dubokog učenja (engl. *deep learning*). U slučaju detekcije objekata potrebno je prikupiti podatke s više kamera u stvarnom svijetu i zapisivati ih na disk te potom u laboratoriju vršiti provjeru algoritama na prikupljenim podacima. Ovaj rad upravo se bavi omogućavanjem testiranja ADAS algoritama u laboratorijima pri tome koristeći AMV Grabber uređaj.

Skupljanje podataka vrši se preko AMV Grabber uređaja te se potom video podaci skladište na računalo i naposljetku dostavljaju do ADAS platforme. U ovom radu obrađena je tema dostavljanja video sadržaja s PC-a na AMV Grabber uređaj, odnosno obrađena je tema programske podrške za reprodukciju video sadržaja za verifikaciju ADAS algoritama na strani PC računala, što uključuje i podržavanja kontrolnih poruka, čitanje video sadržaja s diska kao i njegovo slanje preko PCIe sabirnice u stvarnom vremenu (30 slika u sekundi) za podesiv broj kamera (do 9 kamera).

U drugom poglavlju detaljno je obrađena tematika ADAS sustava, funkcija, arhitektura i smisao AMV ADAS Grabber uređaja te ostala problematika koja je bitna za realizaciju rješenja problema diplomskog rada. Treće poglavlje daje uvid u rješenje problema diplomskog rada i u njemu je detaljno opisan pristup problemu kao i njegovo rješenje. U četvrtom poglavlju dani su rezultati testiranja programskog rješenja. Naposljetku, u petom poglavlju, nalazi se zaključak dobiven nakon rješenja dane problematike i testiranja vlastitog programskog rješenja.

2. AUTONOMNA VOŽNJA I NAČINI PRIKUPLJANJA PODATAKA ZA TESTIRANJE ADAS ALGORITAMA

2.1 ADAS sustavi

ADAS sustavi u povijesti automobilske industrije predstavljaju veliki preokret u načinu pristupanja vožnji, sigurnosti na cestama, smanjivanju ljudske pogreške na minimum, olakšanju svakodnevnice i brojnim drugim aspektima. Dizajnirani su s ciljem omogućavanja visoke razine autonomnosti koje vozilo ima, pa tako korisnik autonomnog vozila tijekom vožnje ne mora sudjelovati u istoj, nego može prepustiti sve autonomnom vozilu koje će odraditi sve ono što vozač u automobilu koje nije autonomno mora uraditi. Brojne su prednosti ADAS sustava, od kojih se prije svega izdvaja sigurnost. Potencijal kojeg autonomna vozila imaju za spašavanje života je značajan, posebice ako se u obzir uzme da, prema istraživanju NHTSA-e, 94% teških prometnih nesreća uzrokuje ljudska pogreška [1]. Vozila temeljena na ADAS sustavima u mogućnosti su ukloniti ljudske pogreške, što znači da povećanje broja autonomnih vozila povećava i razinu zaštite u prometu. Uz sigurnost, ADAS sustavi pružaju dodatne ekonomske i društvene pogodnosti, tako da je istraživanje NHTSA-e (*The National Highway Traffic Safety Administration*) pokazalo da su prometne nesreće u 2010. godini prouzročile stotine milijardi dolara štete, a smanjenje prometnih nesreća svakako bi dovelo do smanjenja troškova nastalih uslijed istih. Prema tome, autonomna vozila bi mogla doprinijeti na način da prometne gužve budu manje, što znači da bi se i vrijeme i novac potrošen na putovanje mogao puno bolje iskoristiti. 2013. godine definirana je klasifikacija razina autonomnih cestovnih motornih vozila te se danas primjenjuje u cijelom svijetu [1]:

1. Razina 0 – apsolutno neautonomna razina u kojoj je korisnik vozila, odnosno vozač, jedini koji u potpunosti i samostalno upravlja vozilom na način da kontrolira upravljanje kočnicama, upravljačem, papučicom gasa i slično.
1. Razina 1 – na ovoj razini postoji djelomična automatizacija određenih parametara vožnje, što uključuje primjerice automatizaciju elektronskog pomoćnog sustava kočenja, elektronsku kontrolu stabilnosti i slično.
2. Razina 2 – primjer autonomije na ovoj razini je tempomat ili sustav prilagodbe brzine vožnje u kombinaciji sa sustavom praćenja bijele kolničke trake, što znači da je na ovoj razini uključena autonomnost za barem dvije primarne kontrole funkcije vozila.

3. Razina 3 – korisnik vozila u vožnju je uključen na način da pravovremeno djeluje samo u kritičnim situacijama, tako da se vozila na ovoj razini autonomije u potpunosti samostalna dok ne dođe do kritičnog trenutka kada je vozač primoran preuzeti potpunu kontrolu nad svim sigurnosnim funkcijama samog vozila.
4. Razina 4 – vozilo je potpuno autonomno te je izrađeno kako bi bilo samostalno i u kritičnim situacijama prateći pri tome uvjete na cesti tijekom cijele vožnje. U ovom slučaju, korisnik autonomnog vozila treba odabrati željenu rutu ili destinaciju, međutim tijekom vožnje mu nisu potrebne sposobnosti za upravljanje vozilom koje je u mogućnosti prometovati i bez prisutnosti vozača.

Kako bi vožnja bila udobnija i sigurnija, autonomna vozila trebaju imati različite senzore pomoću kojih automobilske računalne sustave dobivaju informacije iz okoline. S ciljem omogućavanja ranije navedenih razina autonomije na temelju informacija iz okoline, razvijeni su brojni algoritmi koji će to omogućiti, kao što su neki već ranije navedeni. Značajan dio ADAS algoritama temelji se na informacijama koje su prikupljene s kamere koja se nalazi na prednjoj strani vozila, a prikupljene informacije omogućuju detekciju objekata koja se nalaze ispred vozila. Pomoću informacija s kamere moguće je detektirati razne objekte, kao što su pješaci, prometni znakovi, druga vozila i slično, što znači da takvi algoritmi uključuju i klasifikaciju objekata. Osim toga, postoje i algoritmi koji omogućavaju pomoć vozaču pri parkiranju, korištenjem kamere koje će vozaču omogućiti različite poglede na automobil (npr. pogled odozgor, pogled straga, panoramski prikaz itd.). Budući da korisnik vozila na petoj razini autonomije ne sudjeluje u vožnji uopće, nego samo odabire rutu kojom ga vozilo vozi, potrebni su algoritmi za planiranje kretanja autonomnog vozila, a isti pri tome koriste razne senzore, kao što su GPS (engl. *Global Positioning System*), LIDAR (engl. *Light Detection and Ranging*) itd.

Primjer algoritma koji omogućuje povećanje sigurnosti tijekom vožnje je procjena brzine vozila na temelju snimke kamere koja zamjenjuje retrovizor. Njegova svrha je alarmiranje vozača na potencijalne sudare zbog prevelike brzine vozila odostraga, donošenje odluka o preostrojanju u traku pored i slično. Osim ovakvog algoritma, sigurnost tijekom vožnje postaje veća implementacijom algoritma koji prati stanje vozača pomoću kamere, tako da se može pratiti budnost vozača, njegova pozornost tijekom kritičnih situacija i slično [2].

2.2 AMV ADAS GRABBER

Sve veća složenost ADAS algoritama zahtjeva uzastopna testiranja i davanje potvrda o važećim i nedvosmislenim odlukama koje algoritam donosi. Testiranja predstavljaju vrlo složen proces, budući da je potrebno nakon svake promjene algoritma istu ispitati direktno na poligonu ili u realnim uvjetima u prometu. Kada se sagleda taj scenarij, jasno je kako to iziskuje velike troškove, bilo da se radi o cijeni poligona i ispitivanja algoritma na tom poligonu, ili da se radi o nemogućnosti ispitivanja samog algoritma u realnim uvjetima uzimajući u obzir sigurnost ostalih sudionika u prometu. Obzirom na spomenute probleme, javlja se potreba za projektiranjem ispitnog sustava kojim bi se sustavi za računalnu percepciju okoline vozila mogli testirati u zatvorenim prostorima, odnosno namjenskim laboratorijima. Primjer platforme koja ima tu namjenu je AMV ADAS Grabber koji predstavlja više-kanalni video uređaj za simultano preuzimanje i reprodukciju video sadržaja. To je uređaj koji omogućuje prikupljanje, a osim toga i reprodukciju video signala pri tome koristeći devet kamera prostorno raspoređenih na vozilu. Prijenos podataka se odvija velikim brzinama budući da koristi PCIe (engl. *Peripheral Component Interconnect Express*) sučelje, a omogućeno je i upravljanje samim uređajem [3].

AMV ADAS Grabber uređaj izveden je u obliku kartice koju je moguće utaknuti u standardno računalo. Prijenos podataka odvija se preko PCIe (engl. *Peripheral Component Interconnect Express*) sučelja te je omogućeno i upravljanje samim uređajem. Na slici 2.1 prikazan je izgled AMV ADAS Grabber uređaja.



Sl. 2.1 AMV ADAS Grabber [3]

Prikazani uređaj povezuje se na ispitivanu jedinicu (npr. ploču namijenjenu za implementaciju ADAS algoritama) preko koaksijalnih FPD Link III kablova za prijem i reprodukciju video signala. Centrala procesorska jedinica ovog uređaja realizirana je u hibridnoj programabilnoj mreži Zynq XC7Z030, a navedena osim FPGA (engl. *field-programmable gate array*) dijela sadrži i dvojezgreni procesor ARM® Cortex™-A9. Za rad s ovim uređajem potrebno je imati odgovarajuću programsku podršku koja obuhvaća upravljanje radom uređaja preko PCIe sabirnice, ali i korisničku aplikaciju za snimanje i puštanje video sadržaja, kao i druge opcije omogućene u istoj [3].

FPD Link III predstavlja komunikacijski kanal koji je često korišten u aplikacijama s namjenom automatiziranja prijenosa video podataka. Omogućava prijenos HD digitalnih video podataka, a uz to i dvosmjerni kontrolni kanal koji prenosi kontrolne signale između izvora i odredišta, pri tome koristeći I²C protokol. To je protokol koji je dizajniran tako da omogućava komunikaciju više *slave* integriranih krugova s jednim ili više *master* integriranih krugova. Svaka I²C sabirnica sastoji se od dva signala, a to su SCL (engl. *clock signal*) i SDA (signal za podatke) [4].

PCIe sabirnica (engl. *Peripheral Component Interconnect Express*) omogućava komunikaciju raznih komponenti s procesorom. U ovom konkretnom slučaju, koristi se PCIe x4 sabirnica, koja omogućava direktnu povezanost s računalom preko PCIe utora koji se nalazi na matičnoj ploči. To je sabirnica visokih performansi koja omogućuje prijenos podataka velikim brzinama, što je za AMV ADAS Grabber uređaj izrazito važno, budući da se vrši prijenos velike količine podataka, odnosno video sadržaja [5].

Uređaj se sastoji od 9 ulaza i izlaza koji su paralelno povezani, a centralna procesorska jedinica treba biti u mogućnosti primiti podatke sa deserijalizatora i slati ih preko PCIe sabirnice. Deserijalizator funkcionira na način da serijski zapakirane ulazne signale s kamera pretvara u paralelno sučelje koje se dalje odvija putem 12-bitne sabirnice te odvodi u centralnu procesorsku jedinicu. Osim toga, podaci koji dolaze se deserijalizatora mogu se direktno slati prema serijalizatoru (*Loop Thorough* način rada). Pri tome, korišteni serijalizatori jesu DS90UB914Q proizvođača Texas Instruments te DS90UB913Q deserijalizatori istog proizvođača. Podržani tipovi kamera su OV10635 i OV10640 [3].

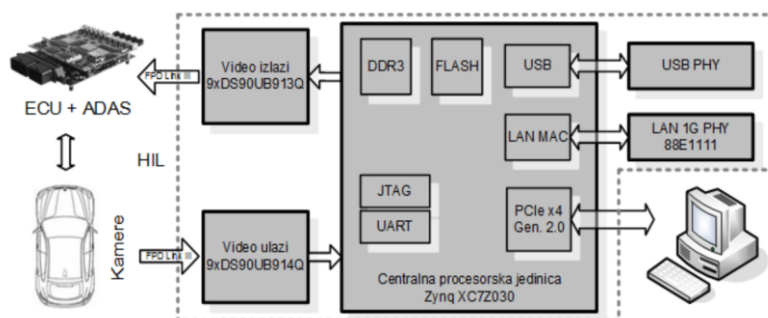
Na slici 2.2 prikazana je topologija AMV ADAS Grabber uređaja, prema kojoj je vidljivo kako AMV ADAS Grabber uređaj treba biti u mogućnosti prenositi podatke na PC preko PCIe

sabirnice, ali ih i slati s PC-a na ADAS uređaj na kojeg se spaja preko odabranog konektora na video izlazu.

Centralna procesorska jedinica koristi sljedeća sučelja:

- PCIe
- UART
- USB
- SD Card
- 1Gb Ethernet
- 1GB DDR3

Na slici 2.2 prikazana je topologija AMV ADAS Grabber uređaja.

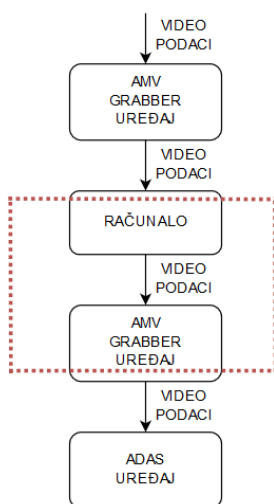


Sl. 2.2 Topologija AMV ADAS Grabber uređaja [3]

Namjena AMV ADAS Grabber je dvostruka: prikupljanje video podataka korištenjem navedenih kamera te slanje na računalo i spremanje na tvrdi disk samog računala. Ta putanja odnosi se na proces snimanja. Nakon skladištenja video podataka u RAW formatu, oni se mogu konvertirati u JPEG, odnosno MPEG format kako bi korisnik mogao pregledati pomoću standardnih preglednika slika ili odabranog video *playera*. Nakon skladištenja na tvrdi disk računala, podaci se dostavljaju do ADAS platforme na kojoj se izvršava algoritam.

U ovom radu obrađena je tematika prijenosa video podataka s računala na AMV Grabber uređaj, odnosno izrađena je programska podrška za reprodukciju video sadržaja na strani PC računala koje komunicira i dostavlja sadržaj namjenskom uređaju za reprodukciju, tj. AMV Grabber ploči. Kako bi se navedena operacija izvršila, programsko rješenje podrazumijeva

čitanje video sadržaja s diska, podržavanje kontrolnih poruka te naposljetku slanje preko PCIe sabirnice. Dio putanje koji predstavlja temu ovog rada na slici 2.3 uokviren je crvenom bojom.



Sl. 2.3 Putanja prijenosa podataka od početne točke-AMV Grabber uređaj do konačne-ADAS uređaj

2.3 Preuvjeti za dostavljanje sadržaja AMV ADAS Grabber uređaju

Dostavljanje sadržaja namjenskom uređaju za reprodukciju podrazumijeva da su video podaci prije slanja uskladišteni na tvrdom disku računala, što znači da treba biti omogućen proces snimanja. Kako bi se taj proces omogućio, potrebno je imati razvijeno programsko rješenje na tri razine:

- Procesor
- Upravljački program (engl. *driver*)
- Korisnička aplikacija

Na prvoj razini programsko rješenje izvodi se programiranje PL-a (engl. *Programmable Logic*) i PS-a (engl. *Processing System*). Što se tiče PL-a, rješenje se izvodi u Vivado Design Suite softveru koji služi za sintezu i analizu HDL (engl. *Hardware Description Language*) dizajnova, te također omogućuje razvoja SoC-a (*System on Chip*).

Programsko rješenje problema na razini *driver*-a, odnosno upravljačkog programa podrazumijeva omogućavanje komunikacije između aplikacije i nižih slojeva, kao i omogućavanje prijenosa samih podataka.

Korisnička aplikacija prije svega treba korisniku omogućiti prijenos podataka s AMV Grabber uređaja na PC kako bi postojala mogućnost slanja u suprotnom smjeru – s PC-a na AMV Grabber uređaj.

Bitno je naglasiti da tematika ovog diplomskog rada nije vezana za razvoj programskog rješenja na spomenutim razinama, nego su to preduvjeti bez kojih slanje podataka s PC-a na AMV Grabber uređaj ne bi bilo moguće.

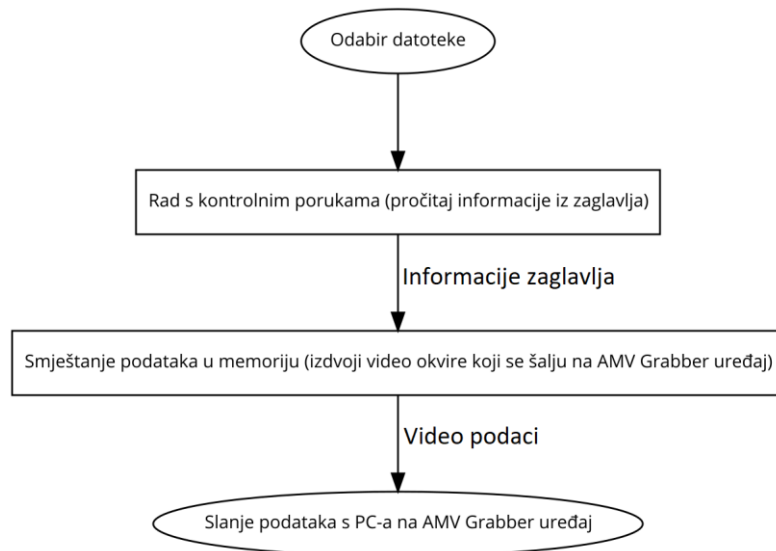
3. OPIS PROGRAMSKOG RJEŠENJA

3.1 Koncept rješenja

Kako bi se ADAS algoritmi mogli testirati na video podacima prikupljenim korištenjem AMV Grabber uređaja, potrebno je prikupljene podatke prvo skladištiti u memoriju računala, a potom ih slati na namjenski uređaj, odnosno AMV Grabber uređaj te poslije toga prema ADAS uređaju kako bi se na poslanim video podacima mogli testirati ADAS algoritmi (prijenos podataka ilustriran je na slici 2.3). Prilikom prijenosa podataka s računala na AMV Grabber ploču, potrebno je omogućiti čitanje iz memorije PC-a, iščitavanje podataka iz datoteke koja sadrži video podatke u RAW obliku, kao i razne kontrolne poruke, prijenos preko PCIe sabirnice te sinkronizaciju kako bi sustav pravilno funkcionirao. Proces prijenosa podataka s PC-a na AMV ADAS Grabber uređaj mogao bi se opisati kroz sljedeće korake:

1. Čitanje iz datoteke
2. Rad s kontrolnim porukama
3. Smještanje u memoriju
4. Prebacivanje podataka iz memorije PC-a u memoriju ploče korištenjem DMA (engl. *Direct Memory Access*)

Dijagram prijenosa podataka s PC-a na AMV Grabber uređaj dan je na slici 3.1. Za implementaciju rješenja odabrano je Windows radno okruženje i programski jezik C.



Sl. 3.1 Prijenos podataka iz memorije računala na memoriju AMV Grabber ploče

3.2 Čitanje iz datoteke

U slučaju prijena podataka s AMV Grabber uređaja na PC, video podaci skladište se u RAW formatu. To je opće prihvaćen naziv za razne vrste formata koji daju neobrađene, odnosno sirove podatke, onakve kakvima ih bilježi senzor kamere. Ovakav naziv nosi budući da podaci nisu procesirani, odnosno da bi korisnik mogao pregledati snimljeni sadržaj, potrebna je konverzija u neki od formata, kao što su JPEG, BMP, MPEG (ukoliko korisnik želi pogledati snimljeni video) te uporaba nekog od standardnih preglednika za sliku ili video. Snimljena datoteka uz video podatke sadrži zaglavlje kao i informacije o svakom pojedinom okviru video zapisa, odnosno zaglavlje video okvira. AMV Grabber uređaj podržava snimanje s više kamera istovremeno, pa ukoliko primjerice koristimo dvije kamere spojene na priključak jedan i dva, video okviri u datoteku se slažu na način prikazan slikom 3.2.

Zaglavlje datoteke	10 video okvira s pripadajućim zaglavljima s priključka 1	10 video okvira s pripadajućim zaglavljima s priključka 2	10 video okvira s pripadajućim zaglavljima s priključka 1	10 video okvira s pripadajućim zaglavljima s priključka 2
--------------------	---	---	---	---

Sl. 3.2 Sastav datoteke koja sadrži video podatke dobivene korištenjem dvije kamere

3.3 Rad s kontrolnim porukama

Datoteka nastala snimanjem video sadržaja, osim video podataka sadrži i kontrolne poruke. Razmjena kontrolnih poruka odvija se putem zaglavlja datoteke i zaglavlja svakog pojedinog okvira. To su dodatne informacije koje služe za lakšu obradu snimljenog video materijala, u smislu prijenosa podataka, kao i pretvorbe istih u neki od standarda za sliku i video. Na slici 3.3 dan je dijagram na kojem je vidljiva struktura datoteke nastale snimanjem, koja je spremljena u memoriju PC-a i koja se treba poslati na AMV Grabber uređaj.



Sl. 3.3 Struktura datoteke koja sadrži video podatke i koja se šalje s PC-a na AMV Grabber uređaj

3.3.1 Zaglavlje datoteke

Na slici 3.4 dan prikazano je zaglavlje datoteke, gdje podaci u crnom okviru predstavljaju informacije o prvoj kameri, dok podaci uokvireni crvenom bojom predstavljaju informacije o drugoj kameri.

Sl. 3.4 Izgled zaglavlja datoteke u kojoj su spremljeni video podaci, prikaz omogućen korištenjem HeX Editora

1. Broj priključka s kojeg snima pojedina kamera
2. Širina slike koju kamera daje
3. Visina slike koju kamera daje
4. Informaciju u broju bajtova po pikselu (dubina boje)
5. Broj okvira u sekundi (engl. *framerate*)

Širina slike koju daje kamera predstavlja dio podatka o rezoluciji kamere, tako da dimenzije mogu biti različitih veličina, ovisno o tipu kamere. Međutim, obje kamere korištene pri izradi ovog projekta daju maksimalnu širinu slike od 1280 piksela. Prema primjeru na slici 3.3 za prvu kameru korištena je rezolucija od 1280 piksela, te je širina slike informacija koja zauzima 12 bita.

Visina slike drugi je podatak koji tvori rezoluciju dane slike, on kao i širina slike ovisi o kameri. Međutim, obje korištene kamere podržavaju više mogućih visina, pa tako imamo uobičajene visine od 1080, 800 i 720 piksela [6]. Kao i širina slike, za zapis informacije o visini koristi 12 bita.

Obzirom na širine i visine snimljenih okvira, vidljivo je kako su (uobičajene) moguće rezolucije sljedeće [6]:

1. 1280 x 1080
2. 1280 x 800
3. 1280 x 720

Broj bajtova po pikselu označava dubinu boje, odnosno način opisivanja koliko informacija o boji sadrži svaki piksel slike, te je to podatak koji se zapisuje koristeći 4 bita.

Broj okvira u sekundi predstavlja broj video okvira koji se prikazuju po sekundi videa. Ovaj podatak, kao i rezolucija okvira je ograničen kamerom te zauzima 1 bajt, a obje kamere podržavaju 30 okvira po sekundi.

Sve ove informacije, kako bi se pravilno iščitale i kako bi se mogle koristiti u daljnjem radu za prijenos, potrebno je obraditi i pohraniti u odgovarajuće strukture. Obzirom na način zapisa pojedine informacije, odabrana je sljedeća struktura za pohranu istih, prikazana u tablici 3.1.

Tablica 3.1 Opis strukture podataka za pohranjivanje parametara zaglavlja datoteke

TIP PODATKA	NAZIV VARIJABLE
uint8_t	portNumber
uint8_t	Bpp
uint16_t	Width
uint16_t	Height
uint8_t	frameRate

Zaglavlje datoteke zauzima 1024 bita, odnosno prvi kilobajt datoteke, što je vidljivo u nastavku na slici 3.5. Budući da su informacije iz zaglavlja bitne za daljnji proces slanja video podataka, iste je potrebno pročitati iz niza bitova. Indeks bita pojedine varijable prikazan je u tablici 3.2.

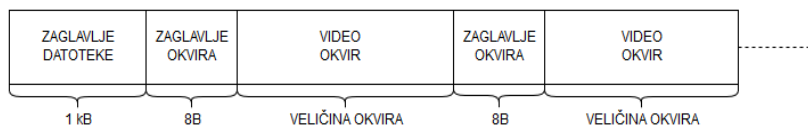
Tablica 3.2 Opis strukture zaglavlja datoteke

NAZIV VARIABLE	BROJ BITOVA	OPIS	INDEX BITA
portNumber	4	Označava broj porta na kojeg je kamera spojena	1-4
Bpp	4	Broj bajtova po pikselu (dubina boje)	4-8
Width	12	Širina video okvira kojeg daje kamera	8-20
Height	12	Visina video okvira kojeg daje kamera	20-32
frameRate	8	Broj okvira u sekundi	32-40

Pojedinom bitu u datoteci (odnosno vrijednostima ranije navedenih varijabli) pristupamo pomoću pokazivača, te uz pomoć raznih logičkih operatora saznaje se vrijednost zapisanu u određenom bitu.

Commented [g1]: zapisana u

Sastav i struktura dobivene datoteke prikazana je na slici 3.5.



Sl. 3.5 Prikaz datoteke uključujući veličine svih dijelova

Kao što je vidljivo na slici 3.5, nakon zaglavlja datoteke nalazi se zaglavlje okvira veličine 8 bajtova, potom podatkovni dio prvog video okvira čija je veličina određena parametrima navedenim u strukturi zaglavlja datoteke.

Formula za izračunavanje veličine video okvira jest:

$$veličina\ okvira = width * height * bpp [B] \quad (3-1)$$

Primjerice, za veličinu okvira snimljenog s kamere koja ima rezoluciju 1280x1080 i dubinu boje od dva bajta dobiva se sljedeći rezultat:

$$veličina\ okvira = 1280 * 1080 * 2 = 2764800\ B$$

Prema veličini pojedinog okvira, u mogućnosti smo pristupiti informacijama zaglavlja video okvira na način da pokazivač pomičemo za iznos koji je jednak veličini video okvira, tako da je to jedna od ključnih informacija za izdvajanje pojedinih okvira.

3.3.2 Zaglavlje okvira

Na slici 3.6 prikazani su podaci koji predstavljaju video podatke, a uokvireni podaci predstavljaju informacije zaglavlja okvira:

```
00 00 00 01 00 00 D0 46 70 0E 30 08 F0 0D 30 08
F0 0E 30 08 60 0E 70 08 30 0E D0 08 F0 0E F0 07
30 0E 60 08 F0 0E 30 08 80 0E 10 08 50 0E 90 08
70 0E 70 08 60 0E 70 08 30 0E 30 08 30 0E F0 07
70 0E 70 09 30 0F 30 08 F0 0D 60 08 B0 0E 70 08
10 0F 90 08 30 0F 70 08 D0 0E 90 09 70 0F 70 08
B0 0E F0 08 10 0F 00 09 30 0F B0 08 B0 0E 60 08
70 0F 50 09 F0 0E E0 08 10 0F 30 09 30 0F 70 08
30 0F E0 08 20 0F E0 08 60 0F A0 09 20 0F 60 09
70 0F 70 09 90 0F 30 09 30 0F F0 09 30 10 30 09
B0 0E 90 08 E0 0E 60 08 70 10 B0 08 C0 10 C0 09
60 10 F0 08 30 11 90 09 D0 11 E0 09 A0 11 20 0A
B0 12 B0 0A B0 13 70 0A D0 12 D0 0B 50 14 D0 0A
20 15 30 0B 30 15 A0 0C B0 15 70 0C 70 16 E0 0C
D0 17 F0 0C F0 18 E0 0C B0 18 70 0E B0 1B A0 0E
E0 1C D0 0E F0 1B 70 0D 90 1E 70 0F B0 21 30 10
30 22 F0 11 30 23 70 10 D0 26 B0 14 B0 2A F0 13
F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Sl. 3.6 Izgled zaglavlja pojedinog okvira u datoteci koja sadrži video podatke

Zaglavlje svakog pojedinog okvira sadrži sljedeće informacije:

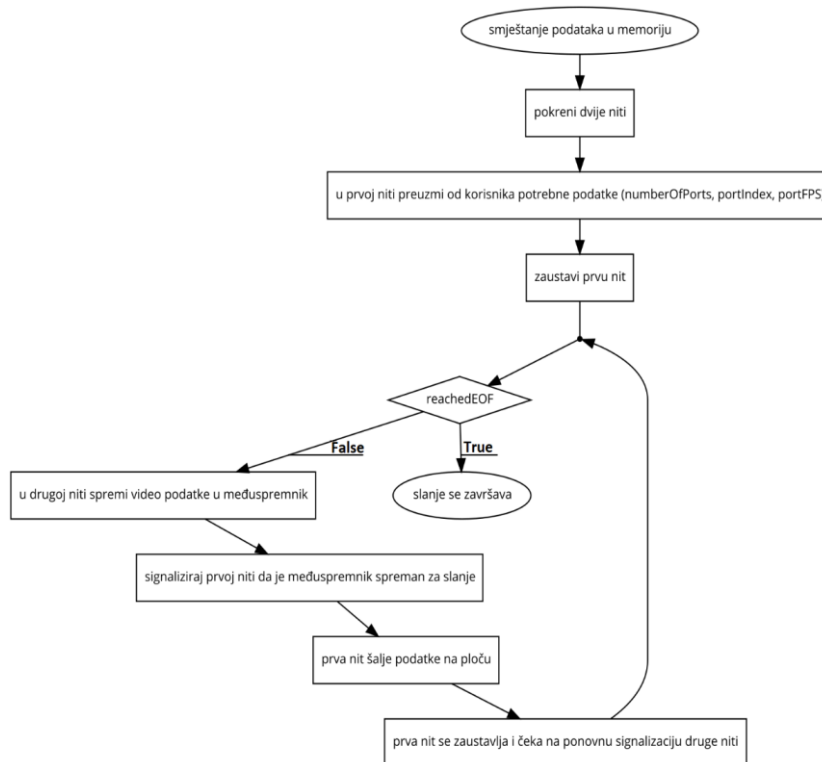
1. Broj porta s kojeg snima kamera koja daje konkretni okvir, zauzimajući pri tome prva 4 bajta u zaglavlju okvira
2. Vremenska oznaka pojedinog okvira, zauzima druga četiri bajta u crnom okviru na slici 3.4

Pri tome, informacija o broju priključka na kojeg je spojena kamera koja daje video okvir omogućuju izdvajanje okvira, što znači da će korisnik imati dodatne opcije prilikom korištenja korisničke aplikacije. Dodatne opcije predstavljaju primjerice slanje video okvira dobivenih s priključka broj dva na izlaz preko priključka broj tri; ili konverzija video podataka s priključka broj jedan u JPEG format.

Vremenska oznaka predstavlja trenutak snimanja pojedinog okvira te se ne koristi prilikom slanja podataka s PC-a na AMV Grabber uređaj.

3.4 Smještanje u memoriju

Nakon završetka obrađivanja podataka koji su spremljeni u datoteci kao što je opisano u potpoglavlju 3.2, započinje slanje podataka koje se odvija kako je prikazano na slici 3.7.



Sl. 3.7 Blok prikaz protokola smještanja podataka u memoriju i prijenosa na AMV Grabber ploču

Aplikacija predstavlja proces unutar operacijskog sustava koji omogućuje raspodjelu računalnih resursa (npr. procesorsko vrijeme, radna memorija itd.). Izvođenje procesa odvija se na način da procesor izvodi određeni niz instrukcija. S ciljem boljeg iskorištavanja resursa, manje složenosti i lakše izgradnje sustava, zadaci unutar jednog procesa dijele se na podzadatke, odnosno niti, a nit predstavlja tok naredbi unutar jednog procesa.

Slanje podataka s PC-a na AMV Grabber ploču je postupak za kojeg je potrebno koristiti niti (engl. *threads*). Prijenos podataka omogućen je korištenjem više niti koje se odvijaju sekvencijalno, a komunikacija između pojedinih niti odvija se pomoću semafora. Semafor

predstavlja varijablu koja kontrolira pristup određenim resursima, na način da se određenim funkcijama kontrolira tijekom izvođenja programa.

Prije svega, kako bi se omogućilo korištenje niti, potrebno je unutar programskog rješenja pozvati funkciju koja omogućuje stvaranje niti te potom unutar niti stvoriti semafore.

Nakon što su niti kreirane, počinje proces slanja podataka. U tu svrhu, korištena je struktura definirana kao što je prikazano u tablici 3.3.

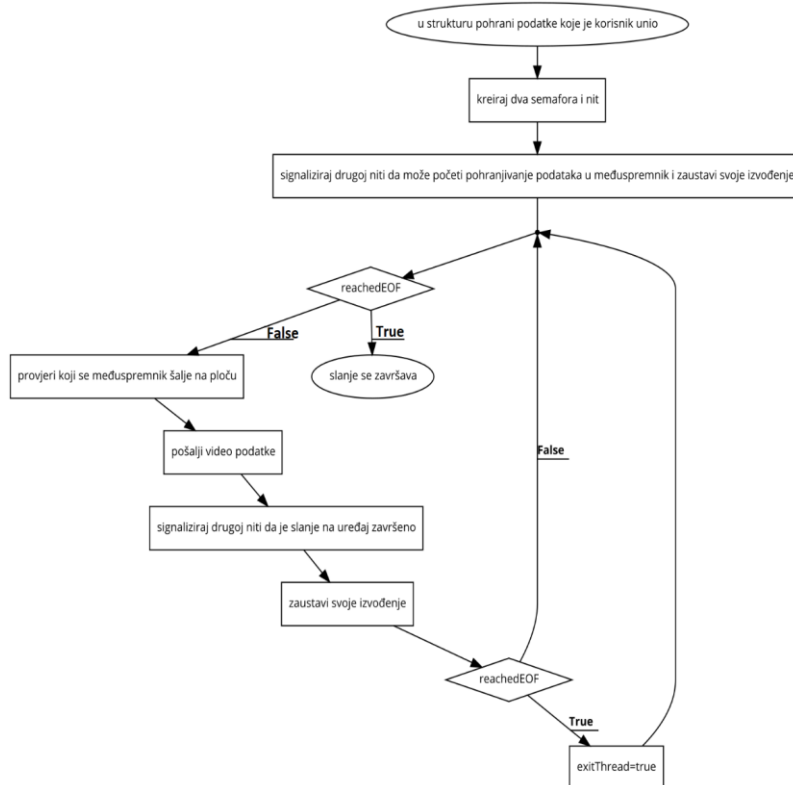
Tablica 3.3 Prikaz korištene strukture za prijenos podataka

Tip podatka	Naziv varijable	Opis varijable
uint8_t	numberOfPorts	Ukupni broj portova s kojih se šalju video okviri
uint8_t	portIndex	Polje koje sadrži indekse portova na koje su spojene kamere čiji video sadržaj prenosimo
uint16_t	portFPS	Broj video okvira s pojedinog porta nakon kojih dolaze video okviri s drugog porta
uint8_t	frameBuffer1	<i>Buffer</i> (hrv. međuspremnik) koji sadrži podatke za slanje
uint8_t	frameBuffer2	<i>Buffer</i> u kojeg se spremaju podaci za slanje
BOOL	usingFirstBuffer	Zastavica za sinkronizaciju između niti
BOOL	reachedEOF	Zastavica za označavanje da je dosegnut kraj odabrane datoteke
BOOL	exitThread	Zastavica koja označava prekid spremanja podataka u međuspremnik
HANDLE	thread1Semaphore	Semafor za komunikaciju između niti
HANDLE	thread2Semaphore	Semafor za komunikaciju između niti

Prije svega, kako bi podaci bili poslani, korisnik aplikacije mora definirati prva tri parametra dane strukture. Nakon toga, stvaraju se niti:

1. Prva nit koja vrši slanje na ploču i stvara drugu nit
2. Druga nit koja vrši izvlačenje video okvira

Na slici 3.8 prikazan je proces izvođenja prve niti.

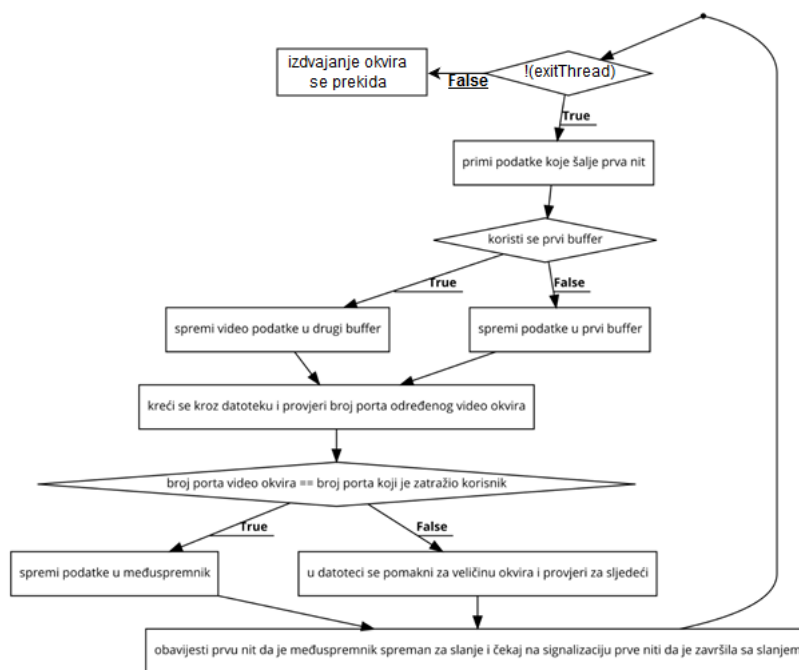


Sl. 3.8 Detaljni blok prikaz izvođenja prve niti

Funkcionalnost koju izvršava prva nit jest samo slanje podataka na ploču. Funkcionira na način da pohranjuje informacije u prve tri varijable strukture prikazane u tablici 3.3. koje joj se prosleđuju od korisnika (*numberOfPorts*, *portIndex*, *portFPS*). Nakon toga, prva nit definira semafore navedene u tablici 3.3: *thread1Semaphore* i *thread2Semaphore* koji će joj omogućiti komunikaciju s drugom niti te također kreira drugu nit. Nakon stvaranja semafora, prva nit signalizira drugoj da može početi sa pohranjivanjem podataka u međuspremnik te zaustavlja

svoje izvođenje do onog trenutka kada joj druga nit signalizira da može započeti slanje podataka na AMV Grabber uređaj. Prije samog slanja, prva nit vrši provjeru zastavice *reachedEOF* (navedena u tablici 3.3) te potom izvršava slanje podataka. Proces pohrane podataka u međuspremnik i slanja podataka je kružan u smislu da se izmjenjuje izvođenje dvije navedene niti sve dok se ne prenesu svi potrebni podaci, a on se zaustavlja u onom trenutku kada zastavica *exitThread* (navedena u tablici 3.3) ne poprimi vrijednost *true*. Ovakvim principom programiranja omogućeno je slijedno izvođenje niti i komunikacija između istih, što znači da se čitanje iz datoteke i slanje pročitanih informacija na AMV Grabber ploču odvija sekvencijalno. Budući da se prenose velike količine podataka, nije moguće alocirati dovoljno memorije kako bi se svi podaci prenijeli odjednom niti sustav ima takve performanse koje bi to omogućile, pa je iz tog razloga odabran ovaj princip programiranja.

Na slici 3.9 prikazan je proces izvođenja druge niti.



Sl. 3.9 Detaljni blok prikaz izvođenja druge niti

Druge nit, koja je aktivirana nakon što se prva nit zaustavi, omogućuje funkcionalnost izdvajanja određenih video okvira. Primjerice, ukoliko odabrana datoteka sadrži video okvire s

portova 1, 2, 3, 4, a korisnik na AMV Grabber ploču želi slati video podatke s kamera koje su spojene na portove 2 i 3, potrebno je napraviti izdvajanje video okvira koje daju kamere na odabranim portovima. Taj postupak se odvija na način da ova nit u jedan od međuspremnik pohranjuje video podatke tako da, krećući se kroz datoteku, za svaki video okvir uzima informacije iz njegovog zaglavlja te na osnovu njih provjerava je li broj porta pojedinog video okvira jednak broju porta one kamere čiji se video okviri šalju na AMV Grabber ploču. Ako su navedene vrijednost jednake, ova nit vrši spremanje video podataka u međuspremnik te obavještava prvu nit da može započeti slanje podataka na AMV Grabber ploču.

Korištenje dva međuspremnik omogućuje sekvencijalni rad između navedenih niti, pri čemu je jedna od bitnih stavki pravilno zauzimanje memorije za svaki od njih. Veličina međuspremnik mora biti dovoljno velika kako bi se mogli spremati svi potrebni podaci i kako ne bi došlo do pisanja/čitavanja izvan zauzete memorije, što znači da veličina međuspremnik ovisi o veličini pojedinog video okvira. Također, dva međuspremnik omogućuju da, dok se u jedan piše, sadržaj drugog se može slati na ploču. Zastavica *reachedEOF* je dodatni parametar koji omogućuje komunikaciju između niti kako bi se omogućilo obavještavanje da je dosegnut kraj datoteke. To je *bool* tip podatka koji ima vrijednost *false* ukoliko kraj datoteke nije dosegnut, a onog trenutka kada su u međuspremnik spremljeni svi potrebni podaci, navedena zastavica poprima vrijednost *true*. Kada zastavica *reachedEOF* poprimi vrijednost *true*, prva nit zna da je dosegnut kraj datoteke i da se slanje prekida.

Ovakav protokol prijenosa podataka s PC-a na AMV Grabber uređaj osmišljen je s ciljem izbjegavanja prepisivanja podataka, u smislu da nije moguće da se istovremeno s jedne memorijske lokacije podaci šalju na uređaj, dok se istovremeno na istu memorijsku lokaciju pišu podaci pročitani iz datoteke.

3.5 Prebacivanje podataka iz memorije PC-a u memoriju ploče korištenjem DMA

Nakon što je međuspremnik popunjen potrebnim podacima koji sadrže video okvire, započinje proces slanja na AMV Grabber ploču. Za taj proces koriste se mehanizmi kao što su *scatter-gather* liste, a isti omogućava automatsko izvođenje niza DMA (engl. *Direct Memory Access*) operacija iz ne-susjednih memorijskih blokova u jednoj operaciji. DMA predstavlja upravljačku komponentu koja omogućuje izravan pristup glavnoj memoriji računala za čitanje ili pisanje podataka, a da pri tome sami procesor ne mora sudjelovati u operacijama. Izraz „*scatter*“ se odnosi na mogućnost pisanja na niz ne-susjednih blokova, dok izraz „*gather*“

odnosni na mogućnost čitanja iz više takvih blokova. Kako bi se DMA operacije izvele ispravno, potreban je popis, odnosno lista takozvanih deskriptora. Deskriptori sadrže informacije o izvorišnim adresama (ukazuju na memoriju na kojoj se nalaze podaci koji se šalju) kao i odredišnim adresama (ukazuju na memoriju na koju se spremaju poslani podaci). Same deskriptor liste stvaraju se u driveru i pohranjuju u memoriju *hosta*, odnosno računala kako bi se mogle proslijediti na uređaj, odnosno *card*. Ovakav način prijenosa podataka omogućen je iz razloga što nije moguće u jednom komadu prenijeti cijeli komad memorije, odnosno slanje svih podataka istovremeno nije moguće, što je detaljnije objašnjeno u četvrtom poglavlju.

Zahtjev za prekidom predstavlja mehanizam koji omogućava centralnom procesoru, tj. softveru koji se na njemu nalazi, asinkronu reakciju na neku od promjena koje su se dogodile na hardverskom bloku. Implementacijom prekidnih rutina u softver omogućuje se reakcija na prekid, budući da prekidne rutine služe za signalizaciju ostatku sustava da se određena operacija izvela. Također, prekidne rutine ne obavljaju obradu samog događaja, osim ako nije riječ o nekoj od elementarnih obrada koje ne zahtijevaju značajan utrošak procesorskog vremena [6].

Dva su moguća smjera slanja podataka [7]:

1. H2C (engl. *Host to Card*) – od računala prema uređaju
2. C2H (engl. *Card to Host*) - od uređaja prema računalu

Način prijenosa podataka s računala na uređaj funkcionira na sljedeći način [7]:

1. Pristupanje uređaju i inicijalizacija DMA.
2. Korisnički program omogućuje čitanje iz datoteke, alocira memoriju za međuspremnik u kojeg se pune dani podaci, te prosljeđuje pokazivač na navedeni međuspremnik funkciji za zapisivanje koja je povezana sa specifičnim uređajem (funkcija za zapisivanje prima i veličinu međuspremnika koji se prenosi na AMV Grabber uređaj).
3. Upravljački program stvara deskriptore uzimajući u obzir veličinu podataka koji se šalju te potom inicijalizira DMA koristeći početnu adresu deskriptora
4. Upravljački program upisuje u kontrolni registar određenu vrijednost kako bi omogućio početak prijenosa.
5. DMA čita deskriptore *host*-a i započinje obradu svakog pojedinog deskriptora
6. DMA preuzima podatke koje šalje *host*, te ih prosljeđuje na *card*, odnosno na uređaj. Nakon što su svi podaci poslani, DMA generira zahtjev za prekidom u smjeru računala.

7. Provjerava se postoje li greške u prijenosu kao što se vrši i provjera koliko je točno deskriptora obrađeno.
8. Nakon utvrđivanja da pogreške ne postoje, upravljački program vraća informaciju o tome kolika je količina prenesenih podataka s ciljem da i softver na korisničkoj strani može provjeriti jesu li podaci upisani u međuspremnik koji se šalje ispravno poslani.

4. TESTIRANJE PROGRAMSKE PODRŠKE ZA REPRODUKCIJU VIDEO SADRŽAJA ZA VERIFIKACIJU ADAS ALGORITAMA NA STRANI PC RAČUNALA

Za potrebe testiranja vlastitog programskog rješenja odrađena su dva slučaja:

1. Testiranje brzine prijenosa podataka preko PCIe sabirnice obzirom na veličinu poslanog paketa
2. Testiranje gubitka paketa
3. Testiranje brzine čitanja podataka s tvrdog diska

4.1 Testiranje brzine prijenosa podataka preko PCIe sabirnice

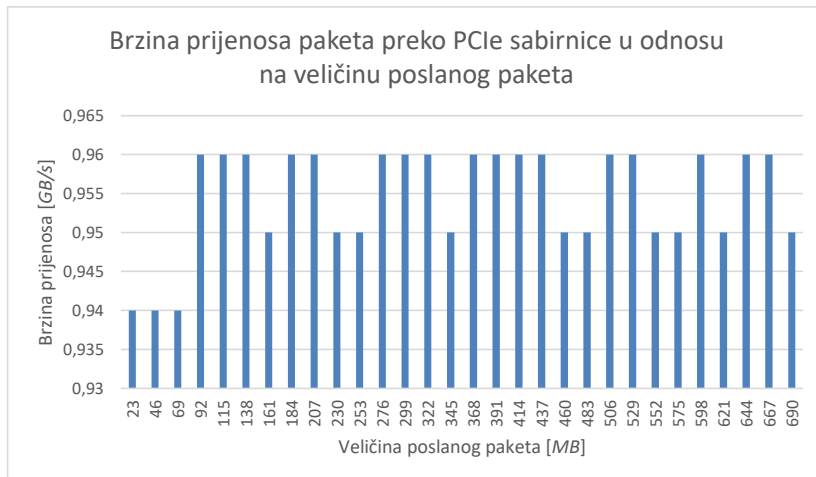
Brzina prijenosa računa se prema izrazu 4-1.

$$brzina\ prijenosa = \frac{\frac{veličina\ poslanog\ paketa\ [MB] \cdot 1024 \cdot 1024}{vrijeme\ prijenosa\ [\mu s] \cdot 10^{-6}}}{10^9} \quad [GB/s]$$

(4-1)

Prema izrazu 4-1, veličinu poslanog paketa dva puta množimo s 1024 da bismo dobili veličinu poslanog paketa u bajtovima te istu potom dijelimo s vremenom prijenosa koje množimo s 10^{-6} budući da je vrijeme prijenosa izraženo u mikrosekundama. Naposljetku, dobiveni iznos dijelimo s 10^9 kako bi brzina prijenosa bila izražena u mjernoj jedinici GB/s .

Podaci dobiveni testiranjem za prvi slučaj dani su u tablici koja se nalazi u prilogu a grafički prikaz dan je na slici 4.1, dok su minimalna i maksimalna vrijednost, kao i srednja vrijednost i standardna devijacija dane u tablici 4.1.



Sl. 4.1 Grafički prikaz rezultata mjerenja brzine prijenosa preko PCIe sabirnice u odnosu na veličinu poslanog paketa

Tablica 4.1 Karakteristične veličine mjerenja brzine prijenosa podataka preko PCIe sabirnice

Minimalna vrijednost brzine prijenosa podataka preko PCIe sabirnice [GB/s]	Maksimalna vrijednost brzine prijenosa podataka preko PCIe sabirnice [GB/s]	Srednja vrijednost	Standardna devijacija
0,94	0,96	0,954667	0,006814

Iz tablice 4.1 vidimo da je minimalna vrijednost brzine prijenosa podataka preko PCIe sabirnice 0,94 GB/s, a maksimalna 0,96 MB/s te da prosječno odstupanje od prosječne brzine prijenosa podataka preko PCIe sabirnice iznosi 0,006814 (standardna devijacija). Prema rezultatima mjerenja prikazanim na slici 4.1, nameće se zaključak kako se brzina prijenosa podataka preko PCIe sabirnice ne mijenja značajno ovisno o veličini paketa te kako ona iznosi približno 1 GB/s. Na osnovu dobivene brzine, zaključuje se kako je omogućen prijenos video podataka prikupljenih sa svih devet kamera rezolucije 1280x1080 koje podržavaju 30 video okvira po sekundi ukoliko ne postoje neke druge prepreke u sustavu.

4.2 Testiranje gubitka paketa

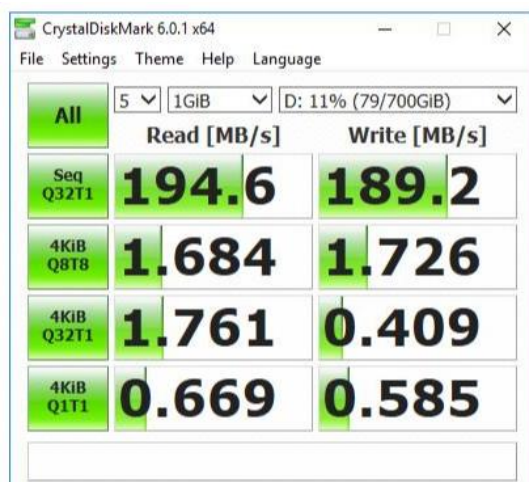
Testiranje gubitka paketa (engl. *packet loss*) koji dostižu na AMV Grabber uređaj s obzirom na veličinu poslanog paketa izvršeno je usporedbom sadržaja poslanih i pristiglih paketa. Pri tome je korišten *TotalComander* softver i uspoređivane su dvije datoteke: poslana, i ona koja je dobivena ispisivanjem memorije sa samog AMV Grabber uređaja, pri tome koristeći *Dump/Restore Data File* alat u *Xilinx SDK* (engl. *Software Development Kit*) softveru. Ovaj test slučaj omogućio je provjeru sigurnosti samog programskog rješenja kako bi se utvrdilo dolazi li do gubitka paketa, odnosno koliko je programsko rješenje ispravno konstruirano.

Pri tome, poslane veličin podataka su bile 100, 200, 300, 400 i 500 MB te je mjerenje za svaku veličinu podataka ponovljeno 30 puta, a samo slanje je izvršeno na način da su se prebacivali nasumični podaci iz RAM-a PC-a u RAM AMV Grabber uređaja.

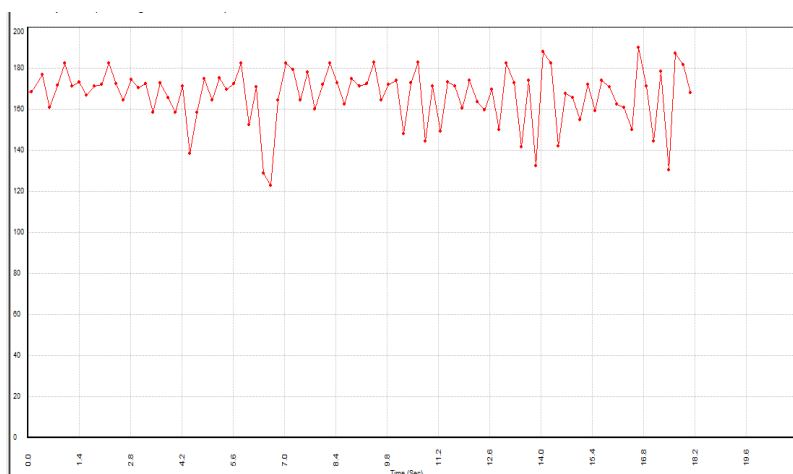
Naposljetku, dokazano je kako je sustav za prijenos podataka s PC računala na AMV Grabber uređaj pravilno konstruiran, te da tijekom prijenosa ne dolazi do gubitka paketa, odnosno sustav je pouzdan što je izrazito bitno za ADAS sustave, jer u njima ne smije doći do gubitka podataka, budući da je sigurnost jedan od najbitnijih aspekata istih.

4.3 Testiranje brzine čitanja podataka s tvrdog diska

Prilikom slanja podataka skladištenih na tvrdom disku, isti se prvo prebacuju u RAM nakon čega se uporabom *scatter-gather* mehanizma vrši samo slanje na AMV Grabber uređaj. Iz tog razloga u sklopu testiranja obavljena je provjera brzine čitanja podataka s tvrdog diska te su podaci za brzinu čitanja s tvrdog diska prikazani na slici 4.2, dok je grafički prikaz dan na slici 4.3. Mjerenje brzine u ovom test-slučaju napravljeno je korištenjem *CrystalDiskMark* softvera.



Sl. 4.2 Rezultati testiranja brzine čitanja podataka s tvrdog diska



Sl. 4.3 Grafički prikaz brzine čitanja podataka s tvrdog diska

Prema danim rezultatima, prosječna brzina čitanja podataka s tvrdog diska je 194.6 MB/s, što je znatno manje u odnosu na brzinu kojom se prebacuju podatci preko PCIe sabirnice (vidljivo na slici 4.1). Na slici 4.2 prikazan je graf na kojem je vidljivo kako se brzina čitanja podataka mijenja tijekom vremena. Varijacije u brzinama čitanja događaju se iz razloga što operacijski sustav zapisivanje izvodi na način da se podaci pišu tamo gdje ima mjesta, bez obzira bili memorijski blokovi susjedni ili ne. Kada memorijski blokovi nisu susjedni, dolazi

do usporavanja brzine pisanja, te analogno vrijedi i za usporavanje brzine čitanja s tvrdog diska što znači da je varijacija brzine čitanja s tvrdog diska uzrokovana smještanjem podataka na nesusjedne memorijske blokove.

Na osnovu dobivenih rezultata iz izvedena tri test slučaja, zaključuje se kako ograničenje sustava predstavlja brzina čitanja podataka s tvrdog diska. Prilikom prijenosa, maksimalna količina podataka koji se prenose u jednoj sekundi je ona veličina koju daju dvije kamere rezolucije 1280x1080 i 30 video okvira po sekundi. To znači da se ne mogu u potpunosti iskoristiti brzine prijenosa koje PCIe sabirnica ima (približno 1 GB/s) jer brzina čitanja podataka s tvrdog diska nije ni približna brzini prijenosa podataka preko PCIe sabirnice. To također znači da je vrijeme prijenosa podataka s PC-a na AMV Grabber uređaj ograničeno brzinom čitanja podataka s tvrdog diska. Također, može se zaključiti kako konstruirani sustav radi ispravno i zadovoljavajuće, a „usko grlo“ sustava je tvrdi disk.

5. ZAKLJUČAK

Tema ovog diplomskog rada podrazumijevala je izradu programske podrške za reprodukciju video sadržaja za verifikaciju ADAS algoritama na strani PC računala. Kroz ovaj rad dan je uvid u ADAS sustave i njihovu ulogu u tehnološkoj budućnosti, kao i sve prednosti istih. Također, opisan je namjenski uređaj AMV Grabber, njegova svrha vezano za izradu diplomskog rada te njegove funkcionalnosti.

Nadalje, u radu je detaljno objašnjen proces slanja podatka s PC računala prema AMV Grabber uređaja, koji obuhvaća četiri cjeline: čitanje iz datoteke, rad s kontrolnim porukama, smještanje u memoriju te naposljetku prebacivanje podataka iz memorije PC-a u memoriju samog uređaja. Čitanje datoteke obuhvatilo je tematiku vezanu za sastav datoteke, te sami početak slanja podataka, dok je u dijelu koji obrađuje rad s kontrolnim porukama dan uvid u zaglavlja datoteke i video okvira, njihova struktura i uporaba tijekom rada. Smještanje u memoriju donosi opis komunikacijskog protokola koji omogućuje sekvencijalni rad dvije niti.

U radu je provedeno testiranje izrađene programske podrške iz čega su utvrđene brzine prijenosa podataka preko PCIe sabirnice, potom je testiran gubitak paketa, kao i brzina čitanja podataka s tvrdog diska. Na osnovu rezultata, zaključak je sljedeći: programsko rješenje omogućuje siguran prijenos podataka bez gubitka paketa, međutim iako su brzine prijenosa podataka preko PCIe sabirnice dovoljne za prijenos video podataka s devet kamera koje pri tome daju 30 okvira po sekundi, ograničenje dobivenog programskog rješenja je vezano za brzinu čitanja podataka s tvrdog diska, koja je znatno manja od brzine prijenosa podataka preko PCIe sabirnice i samim time predstavlja „usko grlo“ sustava.

Obzirom na sve navedeno, nameće se zaključak kako su teorijske osnove kroz dugotrajni rad na ovom diplomskom radu uspješno implementirane te da rezultati vlastitog rješenja ukazuju kako je konstruirani sustav ispravan i da daje zadovoljavajuće (očekivane) rezultate.

LITERATURA

1. D. Williams, "Human Factors Design Guidance for Level 2 and Level 3 Automated Driving Concepts", Washington, DC: National Highway Traffic Safety Administration, kolovoz 2018.
2. A. M Kumar and P. Simon, "Review of Lane Detection and Tracking Algorithms in Advanced Driver Assistance System," *International Journal of Computer Science and Information Technology*, vol. 7, no. 4, kolovoz 2015.
3. Krbanjević M., Rešetar I., Skobić V., "Univerzalna platforma za ispitivanje uređaja za mašinsku vizuelnu percepciju okoline u sistemima za pomoć u vožnji i autonomno kretanje vozila", Zbornik 61. Konferencije za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku, ETRAN 2017., lipanj 2017.
4. Texas Instruments, "I²C over DSP0UB913/4 FPD-Link III with Bidirectional Control Channel", svibanj 2013.
5. Xilinx, dostupno na: (<https://www.xilinx.com/products/technology/pci-express.html>), 10.9.2018.
6. Bjelica M., Teslić N., Mihić V., "Softver u digitalnoj televiziji 1: osnove digitalne televizije i video kodovanja", 1. izdanje, Novi Sad: Fakultet tehničkih nauka, 2017.
7. Xilinx, "DMA/Bridge Subsystem for PCI Express v4.1", Product Guide, travanj 2018.

SAŽETAK

ADAS sustavi, kao i njihovi algoritmi, sve se više razvijaju, a razina autonomije u automobilske industriji raste iz dana u dan. Budući da testiranje samih algoritama iziskuje velike novčane izdatke, javlja se potreba za testiranjem video podataka u laboratorijima. Video podaci skladište se na diskove, te se tijekom testiranja i verifikacije dostavljaju raznim namjenskim uređajima. U okviru ovog zadatka izrađena je programska podrška za reprodukciju video sadržaja na strani PC-a koji komunicira i dostavlja sadržaj namjenskom uređaju za reprodukciju, odnosno AMV Grabber uređaju. Pri tome, realiziran je protokol za komunikaciju preko PCI sabirnice između PC-a i namjenskog uređaja. Također, bitan dio ove teme bila je implementacija kontrolnih poruka, čitanje video sadržaja s diska, kao i njegovo slanje preko PCIe sabirnice u stvarnom vremenu. Rješenje je realizirano u Windows radnom okruženju, u programskom jeziku C.

Ključne riječi: ADAS sustavi, autonomna vozila, testiranje, programska podrška, PCIe sabirnica, kontrolne poruke, Windows radno okruženje, programski jezik C

ABSTRACT

ADAS systems, as well as their algorithms, are increasingly developing, and the level of autonomy in the automotive industry is increasing day by day. Since testing algorithms themselves cost a lot, there is a need for testing video data in laboratories. Video data is stored on disks, and during testing and verification they are delivered to a variety of dedicated devices. The task of this graduate thesis deals with the theme of creating software support for the playback of video content on a PC computer that communicates and delivers content to the AMV Grabber device. In doing so, it is necessary to realize a protocol for communication via PCIe bus between the PC and the dedicated device. Also, an important part of this topic is implementation of control messages, reading video content from a disc, and sending it over PCIe bus in real-time. The solution needs to be implemented in the Windows operating environment, in the programming language C.

Key words: ADAS systems, autonomous vehicles, testing, software support, PCIe bus, control messages, Windows operating environment, programming language C

ŽIVOTOPIS

Ivana Škorić rođena je 9. listopada 1994. godine u Livnu, BiH. Nakon završene Osnovne škole Ivana Mažuranića u Tomislavgradu, upisuje Opću gimnaziju Marka Marulića u Tomislavgradu. Godine 2013. upisala je sveučilišni preddiplomski studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. 2016. godine završava preddiplomski i upisuje diplomski studij na istom fakultetu, izborni blok Mrežne tehnologije. 2017. godine postaje stipendistica na Institutu RT-RK d.o.o gdje obavlja praksu i pri tome radi na diplomskom radu.

PRILOZI

Tablica-rezultati mjerenja brzine prijenosa podataka preko PCIe sabirnice

Veličina [MB]	Vrijeme [μ s]	Brzina [GB/s]
23	253756	0.94
46	51350	0.94
69	76841	0.94
92	100613	0.96
115	125633	0.96
138	149827	0.96
161	178523	0.95
184	200487	0.96
207	223902	0.96
230	251356	0.95
253	277749	0.95
276	299769	0.96
299	325658	0.96
322	351645	0.96
345	382123	0.95
368	394112	0.96
391	427357	0.96
414	445112	0.97
437	475258	0.96
460	498458	0.95
483	522358	0.95
506	532732	0.96
529	577357	0.96
552	598654	0.95
575	620852	0.95
598	641749	0.96
621	631926	0.95

644	676349	0.96
667	715632	0.96
690	763258	0.95